Multimodal Emotion Identification Using Convolutional Neural Networks

Hayk Khachatryan
BS in Data Science
American University of Armenia
Yerevan, Armenia
hayk_khachatryan@edu.aua.am

Vahagn Tovmasyan
BS in Data Science
American University of Armenia
Yerevan, Armenia
vahagn_tovmasyan@edu.aua.am

Supervisor: Anna Tshngryan
Data Science
SkillLab
Yerevan, Armenia
anna.tshngryan@gmail.com

***Abstract*—Our project is a multimodal emotion identification system that can accurately detect human emotions based on facial expressions and voice, utilizing deep learning techniques, specifically convolutional neural networks (CNNs). The goal of our project is to accurately predict 7 main human emotions (angry, disgusted, fearful, happy, neutral, sad, surprised) from facial expressions captured in images and from voice with the help of recordings. Our system mainly uses the TensorFlow Python library for the preprocessing of the data (both images and voice recordings), building and training the neural network systems, and evaluating their final performance.**

In the final stage, mixing the Python libraries with the JavaScript libraries, we added the camera and voice recording functions to make the models work in a live performance. We have also integrated the models in both ways, simple and hybrid. In the simple integration facial emotion detection and voice recognition models work independently, however, in the hybrid one they are connected and the outcome of both models align or not.

I. Introduction

Emotion recognition is still one of the most challenging fields of artificial intelligence, connecting psychological understanding with human-computer interaction. With the help of the detection of emotion from facial expressions and voice intonations, this system can be used in different fields of business, for example, interactive media, healthcare, and customer service. Before, the traditional way of detecting emotions relied on only one factor, either facial expressions or voice. However, as human emotions are complex and facial emotions are connected with voice intonations, the traditional detection of emotions is not so efficient.

The integration of deep learning, particularly convolutional neural networks (CNNs), has developed this field by increasing the accuracy and efficiency of these recognition systems. In contrast with conventional algorithms, deep learning models can make predictions with higher accuracy. Our project uses the deep learning model to develop a multimodal emotion recognition system that will predict the human main 7 emotions with facial and vocal indicators. Mainly, using the TensorFlow library, the system simulates a more realistic and dynamic way of understanding human expressions.

Moreover, to increase the demand for real-time processing in applications such as live customer support, our system incorporates live data capturing through integrated camera and microphone functionalities. With the help of these functions, our project enhances its applicability in real-world scenarios. By implementing both simple and hybrid models, our system aims to achieve higher reliability and accuracy in emotion detection.

This paper will detail the methodologies employed in the system, providing a comprehensive overview of our innovative approach to multimodal emotion recognition.

II. System Overview

To create the multimodal emotion recognition system, we have completed the following essential and mandatory stages: data collection, data preprocessing, model training, integration, and evaluation. All of the mentioned stages are very important not only for reaching the goal of having accurate and efficient emotion detection but also for using it in real-time scenarios and implementing it in different fields.

A. Data Collection

Our data consists of two main components: facial images and voice recordings. Both datasets were taken from Kaggle. Both datasets for training are quite big, which helps the models have higher accuracy. Besides that, the datasets are separated by the seven emotions as mentioned above, which makes the process of evaluating more accurate.

B. Data Preprocessing

As we already have the data, the next phase is the preprocessing. Firstly, we resize and normalize the facial images, so that we can have consistency before entering it in our facial emotion recognition model. We transform the voice recordings into mel-frequency cepstral coefficients (MFCCs), as it is a standard practice in speech processing. This step helps us to capture the essential vocal features for emotion analysis. For the data preprocessing stage, we use advanced Python libraries, OpenCV for the facial images and LibROSA for audio recordings. These libraries ensure that the given data is ready for the model training.

C. Model Training

The facial emotion recognition and voice detection models are trained in parallel. Both models engage convolutional

neural networks (CNNs), utilising the TensorFlow framework for efficient and effective learning. The training procedure is adjusted through the optimisation of hyperparameters and uses methods like dropout and batch normalisation, which will enhance the model generalisation and prevent our model from overfitting.

D. Evaluation

The next phase, after the training of our models, is their testing and evaluation. To understand their accuracy, we have used several visualization tools, using Python libraries Matplotlib and Seaborn. These visualisations show how accurate and well-trained our models are, also showing the potential problems.

E. Integration

After training, evaluating, and saving our models, the next part integrates the trained models in two ways: simple and hybrid. In the simple integration, each model works independently, and in the end, it gives separate emotion predictions from each model. In hybrid integration, both models are connected with each other, and the final output is the predictions given by each model and whether the outcomes from both models are consistent or not. This step is important because it shows the work of a multimodal system and its work with data which single-modality models cannot do. Besides this, we have added the recording and camera features in the integration part to allow the models to be live-action.

III. Data Extraction and Preprocessing

A. Facial Data Acquisition and Organization

The data for facial emotions for this project is taken from a public dataset available on Kaggle, customised explicitly for emotion recognition tasks. The dataset contains a total of 28,717 images for training and 7,186 test images divided into seven emotional categories: anger, disgust, fear, happiness, neutral, sadness and surprise. The following categories are inseparable parts of the model's training because they show the objective output for the emotion recognition procedure.

The data for facial emotions is organised in different directories for every emotion in training and testing folders, and this arrangement makes easier the process of effective separation and handling of image paths and their compatible documentation using Python scripting.

B. Preprocessing for Model Training

For the preparation of data for training, we use several preprocessing steps which are used for optimal model performance:

Resizing and Normalizing Images: Images are resized to 224x224 pixels, which is a standard dimension for CNN inputs, and also, images are normalised to have a pixel value between 0 and 1. Data Augmentation: We use ImageDataGenerator from Keras to increase the training dataset and strengthen the model robustness by mimicking the visual conditions. Data Generators: For this part, we are set up to feed the images in batches, which is highly important for effective memory usage during model training.

C. Visualization of Preprocessed Data

To show the preprocessing results, we show a batch of images after resizing and normalizing, and visualization not only confirms the proper processing of the data but also shows the variety of facial expressions in the dataset.

Fig. 1. Sample of preprocessed facial expressions from the training dataset, representing each of the seven emotion categories.

This data separation and preprocessing approach guarantees the dataset's preparedness for the following model training and evaluation stages, creating a solid base for developing a robust emotion recognition system.

D. Voice Data Acquisition and Organization

The voice data is taken from Kaggle which originally had 2,930 items in it. To have more diversity in our data and more features, we have added datasets from Crema, Savee, and Tess. The updated dataset consists of 12,162 voice recordings, having seven different emotions in them: fear, anger, sadness, disgust, happiness, neutrality, and surprise. In the table, you can see the distribution of the recordings by their emotions:

The distribution of emotions is balanced which provides a robust dataset for training the emotion recognition model.

E. Voice Signal Processing and Augmentation

The initial stage of voice signal processing involves converting the raw audio data into a format suitable for machine learning models. The process includes the following steps:

Emotion	Count
Fear	1923
Angry	1923
Sad	1923
Disgust	1923
Happy	1923
Neutral	1895
Surprise	652

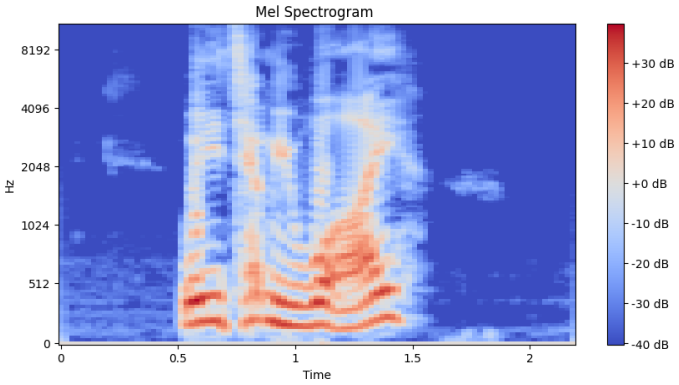

Fig. 2. Mel Spectrogram (This shows the energy distribution across frequency bands over time, providing insights into the temporal dynamics of the speech.)

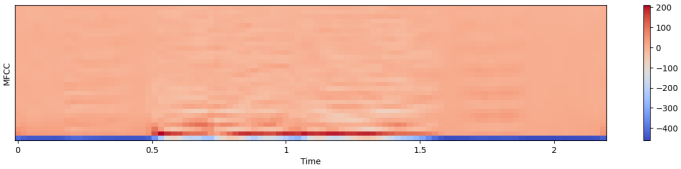

Fig. 3. MFCC (Displays the variation in cepstral features across time, highlighting the texture of the sound that is crucial for recognizing speech and emotion.)

1) Feature Extraction: In this phase, critical features are extracted to capture the essence of the vocal expressions, specifically:

- **Mel-frequency cepstral coefficients (MFCCs)**: These provide the short-term power spectrum of sound, essential for capturing the temporal dynamics of speech.
- **Mel Spectrograms**: These illustrate the spectrum of frequencies of sound as they vary over time, offering a comprehensive view of sound characteristics.

2) Data Augmentation: To enhance the robustness of the model under various auditory conditions, we apply several augmentation techniques:

- **Noise Addition**: Random noise is added to the original audio data to simulate real-world noisy environments.
- **Time Stretching**: Audio files are either stretched or compressed in time to alter their durations without changing the pitch, accommodating temporal variations.
- **Pitch Shifting**: The pitch of the audio is modified to represent different vocal pitches, enriching the dataset's diversity.
- **Time Shifting**: Audio samples are cyclically shifted to vary the starting point of the sound, further increasing the dataset's complexity.

These preprocessing and augmentation techniques are crucial for preparing the data before training the model. They enable the model to recognize emotions from voice under diverse conditions.

F. Visualization of Audio Features

In Figure 2 and Figure 3, you can see the Mel spectrogram and MFCC of a sample audio file to display the preprocessing outputs.

These visualizations help to understand how the audio is processed. Also, it shows that the features are extracted correctly for model training.

G. Data Storage and Setup for Model Training

In the next step, the features of each audio sample are stored in a structured format. Then, we split the dataset into two sets: training and testing. Then we apply one-hot encoding to the emotion labels which prepares them for input into the neural network models.

Finally, the processed data is scaled using standard scaling which normalizes feature values and makes them suitable for efficient model training.

IV. Model Architecture

A. Facial Emotion Recognition Model Architecture

The facial emotion recognition architecture is based on the EfficientNetB7, which is a top-performing CNN known for its efficient handling of complex image data and this can be considered a base, adjusted to address the need for emotion recognition from facial expressions.

The model is structured as follows:

Base Model (EfficientNetB7)

Used for complex architecture able to separate subtle features from images, the model starts with the in-advance trained ImageNet weights to use the benefits of transfer learning, which will be a catalyser for the training process because it starts from a knowledgable base.

Batch Normalization

After the base mode, the batch normalization layer, which normalizes the activations from the previous layer to have a more stable learning process by keeping the mean output value close to zero and the output standard deviation close to zero, is added.

Dense Layer

A layer with 256 neurons follows, which features a regularisation process to reduce overfitting. L2 and L1 regularisations are used, where L2 penalises the square magnitudes of the parameters to have smooth weights and biases to avoid extreme values, and L1 is applied to have a sparsity in the learned weights, which will help to have a less complex and more straightforward interpretable model, which will prevent overfitting.

Dropout

A dropout layer is used to prevent overfitting with a ratio of 45%. During this step, every input unit has a 45% chance of being set to 0 (temporarily), and it will not participate in this step during training, which will help the model reduce the likelihood of relying on coincidental relations.

Output Layer

The last layer has seven neurons, equal to the number of emotions that we have in our given dataset. In this case we have multi-class classification task and the output is the probabilities of the input being in each category, so it uses softmax activation function.

The model is compiled with the extension of Adam optimizer, Adamax optimizer, which is used for noisy gradient datasets or sparse data. Categorical crossentropy which is a loss function is used mainly for multi-class classification problems and it measures the difference between actual distribution and predicted probability distribution. The goal for categorical crossentropy is to minimize this loss, and improve the accuracy for classifying the inputs into their respective classes.

Above is a summary of the compiled model:

```
Downloading data from https://storage.googleapis.com/keras-applications/efficientnetb7_notop.h5
258076736/258076736 [==============================] - 1s 0us/step
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 efficientnetb7 (Functional  (None, 2560)              64097687
 )

 batch_normalization (Batch  (None, 2560)              10240
 Normalization)

 dense (Dense)               (None, 256)               655616

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 7)                 1799

=================================================================
Total params: 64765342 (247.06 MB)
Trainable params: 64449495 (245.86 MB)
Non-trainable params: 315847 (1.20 MB)
_________________________________________________________________
```

The goal of this kind of architecture is increasing the efficiency of the model interacting with the complexity of emotional recognition through deep learning techniques which are optimized for such tasks by improving the accuracy of the crucial indexes. By using both EfficientNetB7 and proper regularization in addition to dropout, we ensure, first of all, that the model performs well, remains robust against overfitting and is compatible with rather variable input data. This makes it an excellent choice for implementation in the real world.

B. Voice Recognition Model Architecture

The voice emotion recognition model is developed to efficiently process the audio data and accurately predict different emotions in voice recordings. To create the model, we used a combination of convolutional neural network (CNN) and long short-term memory (LSTM) layers. However, the main structure is based on deep CNN layers as their effectiveness is higher in capturing the temporal and spectral features of sound data.

DETAILED ARCHITECTURE

The model is composed of several layers designed to process and analyze voice recordings effectively:

Input and Reshape

First of all, the voice recordings are reshaped, so that they can fit the model's input requirements.

Convolutional Layers

The model uses several layers, called convolutional layers, each with different settings, to analyze the sound data. These layers help to understand important features of the sound that are useful for recognizing emotions.After each of these layers, there's a process called batch normalization. This process helps to make the model's calculations more consistent, which speeds up how quickly it learns and makes it more stable as it improves.

Max Pooling Layers

The max pooling layers are added between the convolutional layers to simplify the amount of data the model processes. These layers help the model to focus on the most important features in the sound data, which are crucial for predicting emotions accurately.

Dropout Layers

To ensure that the model doesn't face the overfitting problem (a problem when the model just memorizes the training data), dropout layers are used. During the training process, the dropout layers randomly ignore some of the data paths in the model which helps the model become better.

Flatten and Dense Layers

After the model has processed the data through all layers, the data is transformed into a single long list (flattened) and then processed by a dense layer. This dense layer, consisting of 512 neurons and using a function called ReLU, helps the model make complex decisions from the learned features. The final part of the model is a softmax layer with 7 outputs, one for each emotion category. This layer calculates the likelihood of each emotion being the correct response based on the learned features.

Compilation and Training

The next step for our model is using the Adam optimizer. This optimizer is very effectively handling the complex data during the learning process, especially when the data is audio data. The Adam optimizer uses categorical crossentropy error measurement. This error measurement type is useful when there are different categories to choose from. Besides the mentioned, we also add some smart training features to our model, so that it trains effectively.

Model Checkpointing: This feature saves the best version of the model that performs on our validations tests. With the help of this, we won't use the best version of the model, even if the model's performance goes down in the next training rounds.

Early Stopping: The Early Stopping feature stops the training when the model starts to perform worse on the validation tests. With the help of this feature, we avoid unnecessary training, save time and resources.

Learning Rate Reduction: This feature reduces the rate of the model's learning of new things, when the model's improvement starts to slow down.

Model Summary

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 2376, 512)         3072

 batch_normalization (Batch  (None, 2376, 512)         2048
 Normalization)

 max_pooling1d (MaxPooling1  (None, 1188, 512)         0
 D)

 conv1d_1 (Conv1D)           (None, 1188, 512)         1311232

 batch_normalization_1 (Bat  (None, 1188, 512)         2048
 chNormalization)

 max_pooling1d_1 (MaxPoolin  (None, 594, 512)          0
 g1D)

 dropout (Dropout)           (None, 594, 512)          0

 conv1d_2 (Conv1D)           (None, 594, 256)          655616

 batch_normalization_2 (Bat  (None, 594, 256)          1024
...
Total params: 7193223 (27.44 MB)
Trainable params: 7188871 (27.42 MB)
Non-trainable params: 4352 (17.00 KB)
_________________________________________________________________
```

Fig. 4. Voice emotion recognition model summary

This model is trained in a way that it can handle the complex audio signals. Besides that, it is accurate enough to understand different nuances in vocal expressions which makes the model highly effective for real-time emotion detection.

V. Training and Evaluation

Face recognition

The trained model utilized a series of epochs to gradually learn to disentangle facial expressions from each other and classify them among the 7 states of defined emotions. Through the usage of a mixture of train and validation data which represents the heterogenized data, we expected that the model will generalize well on the new (unseen) data by avoiding potential bias and oversight of model learning.

Model Performance Evaluation

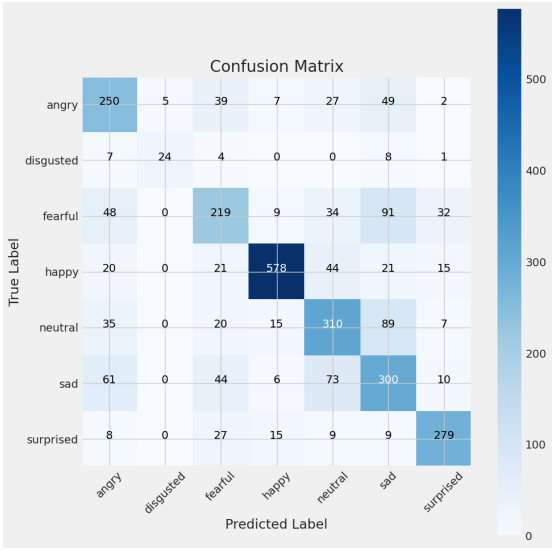

Fig. 5. Confusion Matrix

The confusion matrix provides a definition of the performance of a classification algorithm. Key observations include:

- **High Accuracy for 'Happy'**: The highest number of correct predictions in this model is for the 'happy' emotion with 578 true positives, which indicates that the model consistently performs in recognizing happiness.
- **Confusions Noted**: Not all emotions keep their uniqueness and some of the emotions create confusion, such as 'anger' being confused with 'fear' and 'surprise', which means that the model has difficulty differentiating between similar emotions.
- **Room for Improvement**: 'Disgust' and 'surprised' are found to have the lowest accurate positive rates, which means that the model has a problem in this area and an improvement is needed in selecting the data or applying some changes in the model architecture.

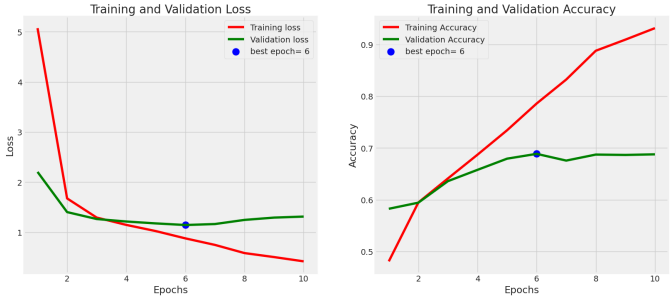

Fig. 6. Training and Validation Loss and Accuracy

The first chart helps detect the training and validation loss and accuracy over the implementation epochs, thus providing valuable information regarding the model learning procedures. The second chart tracks the training and validation loss and accuracy over the epochs, providing insights into the learning dynamics of the model:

- **Loss Reduction**: The training and validation loss starts to decrease, and such a pronounced decrease over time, which means that the model learns well and as much as it can, and in turn, there will be no overfitting danger.
- **Accuracy Improvement**: The precision metric boosts and reaches maximum value with the rise of the training and validation epochs, which is expected because the model learns some data and trends. To format the charts, we will select this technique so that tracing will be easily visualised when the model achieves the highest point on the validation set. This is the optimal balance between learning and generalization until it will continue and there will be overfitting.

Conclusion

To summarize, our project matches well and produces a multimodal emotion recognition system which is efficacious in precisely predicting human emotions based on facial and vocal cues. The system was highly effective in many fields, but some of the specific areas still require further refinement, such as differentiating among facial expressions like ”Disgust” and ”Surprise”, as well as Angry and Sad vocal expressions. Through the extended datasets, adjusted the architectures of the models as well as with the advancement in preprocessing and features extraction methods, we would be able to produce a very effective system. The adoption of such a robust emotion recognition system gives very big to many sphere of real life applications, whether it is a customer service, healthcare, or entertainment industry.

Voice recognition

Training Process: The voice emotion recognition model has processed extensive training over multiple epochs to optimize its ability to predict emotion based on vocal features. The training process included techniques of data augmentation such as adding background noise, altering pitch and speed and applying time-shifting. Using training and testing datasets, the model has a robust assessment of the model's generalization capabilities. The input training dataset refined the model's weight values, while the output validation dataset examined the generalization features of the model after each epoch's performance. This approach helped prevent these models from overfitting and setting up optimal performance on unseen data.

Model Performance Evaluation

From the above confusion matrix, we can interpret that the majority of the emotions are classified correctly, which shows the model is more accurate. However, the model also some misclassifications, for example “Fear” is mistaken for “Surprise” and “Angry” for “Sad”. This shows that the vocal signatures of these emotions overlap, which makes it challenging to differentiate clearly.

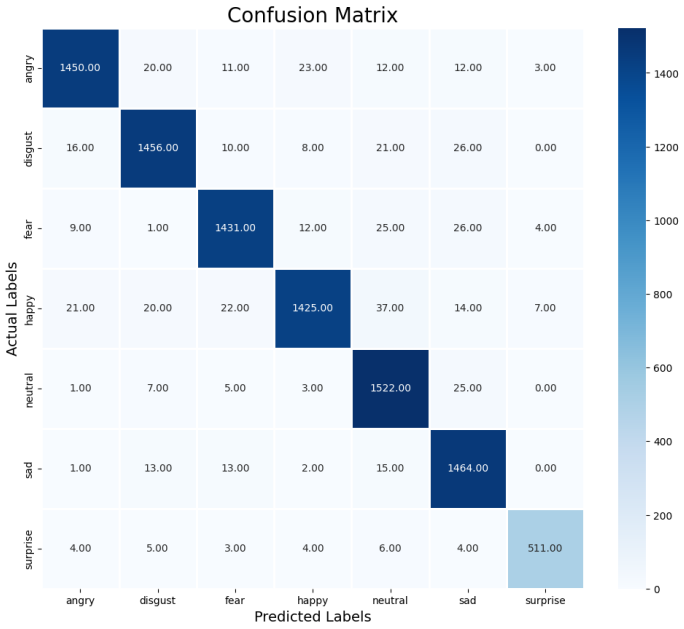

Fig. 7. Confusion Matrix

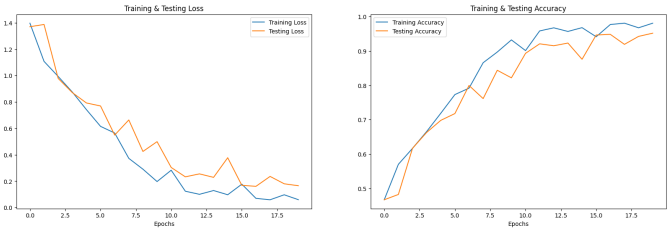

Fig. 8. Confusion Matrix

Figure 8 shows the progression of training and testing loss and accuracy. We can see that both training and testing loss decreases when the epochs increase. This shows that our model is learning effectively, and its accuracy increases during the time. On the other hand, the training and testing accuracy converge which shows that our model does not overfit to the training data.

The above visualizations show that our model has performed quite well. It has higher accuracy for the majority of the emotions, however, misclassifications show that there is room for improvement. These can be more nuanced feature extraction techniques or some additional layers in our model, which will help to better capture the emotions in the voice.

Results and Model Integration

Overview

Our project has successfully predicted both facial and vocal emotions by our trained models. The next step is integrating the models, also adding live camera and voice recording features which will allow us real-time emotion detection. We have done two integrations, simple and hybrid.

Simple Integration

In the simple integration, facial recognition and voice detection models work independently. Photos and audio recordings are processed to their respective models. For the facial recognition part, the system captures the photo by using the camera feature, then it processes the photo, and based on the facial expressions, predicts the emotion. The voice recognition model firstly records the audio, then does the analysis part for the vocal features and predicts the emotion.

Hybrid Integration

The hybrid integration combines both models and does a unified emotion prediction. This integration type uses camera and microphone features by capturing the photo and recording the voice, then processing the facial and vocal inputs separately to their corresponding models. In the end, the system compares the findings of both models to understand the prediction accuracy. For example, if the face recognition model shows that the output is happy and the voice recognition model predicts a positive tone of voice, the system outputs a positive emotional state.

Camera and Voice Recording Features

Now let's understand how the camera and voice recording features work.

Camera Feature: For the camera feature, we used a JavaScript function which captures the photo through the web camera. After capturing the photo, the function flips horizontally and then saves the photo.

Voice Recording Feature: The voice recording feature also uses a JavaScript function which records the audio by the microphone. After recording, the audio file is saved and ready to be processed by the voice emotion recognition model.

Results from the Integrated System

Simple Integration: The simple integration works well in the situations where we need only one model, either facial emotion recognition or voice emotion recognition.

Hybrid Integration: The hybrid integration will be needed when we need to use both models, predicting and comparing the both facial emotions and voice emotions.

Conclusion and Future Work

Conclusion

Our project has developed a multimodal emotion recognition system that successfully predicts human emotions from facial expressions and vocal cues. It not only has a high accuracy of predicting emotions but also includes a real-time emotion detection system with camera and voice recording features. The implementation of simple and hybrid integration gives us different opportunities and ways to use our trained models, each of which has its advantages.

Future Work

Looking ahead, there are several ways for further development and improvement of the emotion recognition system

Enhanced Data Integration: Complex Decision-Making Algorithms: Use smart algorithms in the hybrid approach that can distribute the outputs of the facial and voice models dynamically according to the context or layer of certainty whichever one raises more.

Real-Time Data Synthesis: Mirror uses facial and voice data processing as well as in-time data synthesis to momentarily provide feedback, which might be rather suitable for interactive applications.

Broader Emotional Spectrum: Expand Emotion Categories: At the moment, it identifies only the basic kinds of emotional feelings (only the main 7 emotions). Following updates could include a more extensive scale of emotions, including, perhaps even greater subtlety, within the primary six emotions.

Cultural Sensitivity: Tailor the system to read and decipher emotional expressions from a variety of cultural backgrounds, implementing how cultural variations might affect the type and expression of emotions.

Advanced Model Training Techniques: Incremental Learning: Apply learning methods that prevent the system from forgetting the past information provided while learning more content from new data. This, therefore, enables the system to adapt to changes over time as well as adapt to the context the emotional expression is occurring.

Adversarial Training: Adversarial training implies a way of enhancing the robustness of the models that helps them to withstand inputs that are used to trick or misguide the method.

Application Development: Healthcare Applications: Create applications for intellect monitoring, where the system can help the staff to have more insight into the patients' emotional states by analyzing the data sets mentioned above.

Educational Tools: Take the system to schools where lessons can be customized for the way students feel so that they can participate and be productive more. This could mean that the generations that will come after can learn better.

With the help of this emotion recognition system, we will have one step forward to the integration of AI with human-interfacing technologies. After going deep and developing this idea, these kinds of systems have the potential to change the interactions of humans with machines, making them more intuitive. Future development of these systems will show that they have a positive impact on society.

References

[1] Kalyta, O., Barmak, O., Radiuk, P., & Krak, I. (2023, August 31). Facial emotion recognition for photo and video surveillance based on machine learning and visual analytics. MDPI. Retrieved from https://www.mdpi.com/2076-3417/13/17/9890

[2] Legara, J. S. (2023, June 11). Frame-by-frame: Tracking emotions in videos with ai. Medium. Retrieved from https://medium.com/@johnsolomonlegara/frame-by-frame-tracking-emotions-in-videos-with-ai-ee31a1a05ab6

[3] Li, Y. (n.d.). Deep learning of human emotion recognition in videos. Retrieved from https://www.diva-portal.org/smash/get/diva2:1174434/FULLTEXT01.pdf

[4] Livingstone, S. R. (2019, January 19). Ravdess emotional speech audio. Kaggle. https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio

[5] Ares. (2020, December 11). Emotion detection. Kaggle. Retrieved from https://www.kaggle.com/datasets/ananthu017/emotion-detection-fer

[6] Bhattiprolu, S. (2023, August 23). Bnsreenu/python_for_microscopists. Retrieved from https://github.com/bnsreenu/python_for_microscopists/tree/master